Technical White Paper
Joe Macker: Naval Research Laboratory
Winston Dang: University of Hawaii
Last Revision: April 1996

The Multicast Dissemination Protocol (MDP) version 1 Framework

Abstract
This white paper outlines a simple protocol framework that was developed for reliable multicast dissemination of data files. It also provides some historical development and testing history. The general framework described here was originally developed and used by the Image Multicaster (IMM) application within the Internet Mbone for reliable multicast file transfer. This document describes the more general use of the protocol framework as a reliable bulk file transfer technique. We discuss the present operational modes, some performance issues, and the basic application data units (ADUs) used. This is not intended to be a detailed protocol specification document, but rather a broad description of the basic protocol features and a discussion of issues.

Introduction
The Image Multicaster (IMM) application was originally designed and implemented during 1993 as a reliable multicast tool with the intention of disseminating compressed imagery files to a group of multicast receivers. It was operated and tested within the worldwide Internet Multicast Backbone (Mbone) for reliable bulk transfer of satellite imagery files using UDP/IP multicast transport as defined in RFC 1112 [2]. Besides imagery dissemination support, the IMM reliable multicast framework proved useful and was demonstrated as a general reliable multicast file transfer application. In order to further clarify the general application of this framework, we refer to it as the Multicast Dissemination Protocol (MDP) within this and future documents. This introductory white paper provides a functional overview of the MDP protocol framework and discusses application data unit (ADU) types used in the early implementation.

Motivation
Generic IP multicast builds upon the connectionless, best effort service provided by UDP or raw IP and provides no guaranteed reliable or ordered delivery of data to end applications. Rather than an apparent reliability disadvantage, this feature can be advantageous in supporting a rich set of application classes that can best determine their own definition of reliability and protocol operation by incorporating application layer considerations [5]. While there are a number of application classes with unique requirements motivating a variety of reliable multicasting design approaches, there is a general need for non-real-time bulk file transfer support. This is the specific category of problem to which MDP provides a candidate solution. There are a numerous references for more extensive discussions of reliable multicasting design and application issues [4,7].

Description of IMM/MDP Approach
MDP is a protocol framework that implements reliable multicasting for bulk files using application layer framing concepts [5]. The MDP framework consists of both receiver and source software modules. Multicast receivers wishing to subscribe to a multicast file dissemination service require

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **APR 1996**	2. REPORT TYPE	3. DATES COVERED **00-00-1996 to 00-00-1996**
4. TITLE AND SUBTITLE **The Multicast Dissemination Protocol (MDP) version 1 Framework**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Avenue, SW,Washington,DC,20375**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**11**	

the receiver module, while dissemination sources require the source module functionality. Present distributed versions of the IMM software package provide both receiver and source functionality.

At present, MDP performs its protocol functionality using a single multicast group for an active file transfer session. MDP uses selective negative acknowledgement (NACK) to request repairs of missing data. It uses NACK suppression methods and event timers to reduce congestion and retransmission requests within the network. Both NACKs and retransmissions are multicast to the group to aid in feedback suppression. This is an important scalability feature for operating such a framework within a WAN infrastructure with a potential large receiver set. This feature and additional message aggregation functionality help reduce the likelihood of control message network implosion effects.

History of Network Usage

As mentioned, the IMM/MDP approach has been used within the Mbone for periodic dissemination of satellite imagery to subscribed receivers worldwide. With a simple connection to the Mbone, IMM allowed users from around the world to continuously view the latest weather images from satellites covering most of the earth. The initial project and testing lasted for several months transmitting hourly satellite images derived from the GMS-4, GOES-7, METEOSAT, and GOES-8 satellites. Rapid dissemination of time sensitive information without the need for a worldwide hierarchical distribution system was a key beneficial feature of reliably multicasting over the Mbone.

Protocol Framework and Operation

The following section discusses the overall functional operation and design of the MDP version 1 protocol framework. There are a number of protocol variables and operational modes that are described, but it is not the intent of this document to provide functional specification level of detail. Figure 1 is a high level description of the operation of the MDP framework. Basic UDP/IP multicast provides the transport channel for the service and the MDP reliability process provides reliable delivery of transmitted file data to multicast receivers.

```
                                                         -----------
                  |--MDP Reliability Process--|          | Archive |
                  v                           v          -----------
---------    ----------                  -------------        ^
| file/ |    |  MDP   | -------------> |   MDP       |       |
| bulk  | -> | Source | <------------- | Receivers   |---------|
| data  |    |        |   Multicast    |             |         |
---------    ---------- Dissemination  -------------        v
                          Channel                      --------------
                                                       |    Post    |
                                                       | Processor  |
                                                       | e.g, image |
                                                       |   viewer   |
                                                       --------------
```

Figure 1: High Level MDP Operation

The MDP source fragments file data into a series of data units based upon the MDP maximum data unit (MDU) length setting. This data is provided to receivers over a multicast dissemination channel using UDP/IP multicast transport. Source and receiver MDP processes use an initial

transmission cycle and a successive series of recovery cycles to ensure reliable delivery of file data among the group of multicast receivers. The transmission cycle is the period in which data is first transmitted and a recovery cycle is a period during which data repair requests are serviced and selectively requested data repair packets are retransmitted. An operational description of the transmission and recovery cycles is presented later in this document.

Source Features

The MDP source is designed to transmit a file of any type or an entire directory structure to a multicast receiver group. The user can set the source to reliable transmit all files within a hierarchy once only or continuously. If set continuously, the source can multicast files in a round robin scheduled fashion or optionally transmits updated files in the directory after one complete initial pass through the directory. The source checks for updated files by examining the file timestamp and comparing it to the sources last file sent timestamp. In effect, this optional feature allows for an underlying file update frequency to more directly determine the required transmission frequency.

The present MDP source design is based upon an explicit rate control mechanism. The user or an automated rate control mechanism has the ability to control the transfer rate as well as an additional transmission frequency wait period. The transfer rate pertains to the actual transmission rate of individual packets from the multicast source. The source uniformly distributes packet transmission times based upon this setting and the MDU value. The transmission frequency wait period defines an amount of time to wait before starting the transmission of successive files. A zero setting for this wait period allows for continuous file transfers, whereas a setting of 3600 seconds results in the next file being transmitted one hour after completion of the previous file transmission.
If the source is not finished transmitting a complete file before the expiration of the frequency setting, the source will attempt to complete the full file transmission procedure. Therefore a zero setting results in back to back full file transmissions. In the special case of continuous file transmissions, the source has the ability to stack multiple simultaneous file transmissions and recoveries up to a predefined maximum state cache limit (e.g., 8 files). This overlapping of transmission and recovery cycles results in more effective usage of bandwidth during recovery cycle periods when the source is waiting to collect receiver requests for missing data.

For applications desiring positive acknowledgment of files received from receivers, the source provides an optional operational mode that requests that receivers provide a positive acknowledgement upon final receipt. Again, the receiver responses are designed to be random time delayed over a uniform window to smooth out the flow of packet data proceeding back to the source.

Receiver Features

The MDP multicast receiver application has been designed to offer flexibility in processing each newly received file. As shown in Figure 1, receiving nodes have the individual option of allowing the newly received file to be either archived or post processed through a user selected executable (e.g., jpeg viewer, web client, slideshow application), as shown in Figure 1.

In the present design, the receiver is also capable of tracking several file transmissions simultaneously from a single source as well as from multiple sources. Transmission from multiple independent sources allows for collaborative distribution of files between large audiences. The

receiver properly reassembles file transmissions arriving from multiple independent sources sending on the same multicast address.

Within the present design, receivers are free to drop in or out of a group session. However, upon initiation of a receiver, a group membership packet is sent by the receiver to cordially announce participation in receiving session data. At the present time, such announcements are optionally used by other session receivers to track membership and session statistics.

MDP ADU Packet Types

To provide some background terminology to the reader prior to a detailed discussion of protocol operation we present ADU packet types and a brief description of how they are used by MDP. The MDP framework uses five main types of ADU packets to transfer information.

(1) Identification	Used by source convey file information
(2) Data	Used by the source to transmit file data
(3) Missing Data	Used by receivers to report missing data to source
(4) Command	Used by source to query or trigger receiver responses
(5) Statistic	Used by receiver to report summary statistics

Identification (ID) ADU Packet

The ID ADU packet provides file information to the receivers. It is used to advertise upcoming transmissions and the wait period and also indicate the end of transmission cycles. The ID ADU contains the following type of information.

- protocol version
- file identification number (source unique)
- file size (bytes)
- delay interval between transmissions
- file name
- protocol flags

The file identification number is a source unique assigned number which serves as a reference handle to allow receivers to make specific requests concerning a particular file. The ID packet is transmitted at the beginning of each file transmission, the end of each file transmission, and also at the end of its recovery cycle period. In this way, the ID packet helps synchronize the negative acknowledgment recovery cycle period between all of the receivers.

Data ADU Packet

The Data ADU packet is used by the source to transmit the actual data content of a file. Transmission is performed by dividing each file into fixed length segments, the application layer maximum data unit (MDU). These segments make up the payload within data ADU packets.

The Data ADU contains the following type of information.

- protocol version

- file identification information
- position counter (offset)
- block id (optional to identify FEC boundaries,etc)
- flags
-

The position counter along with the MDU size is used to determine the file offset in bytes from the beginning of the file. It allows the receiver to reconstruct the original file by inserting the data segment into a duplicate file at the same offset. Data packets are multicasted sequentially during initial file transmission. During a recovery cycle, data packet retranmissions are selectively triggered by missing data repair request packets from receivers described below.

Missing Data Repair Request ADU Packet

The Missing Data Repair Request ADU packets are receiver requests for retransmission of repair data packets for a particular file from a particular source. The Data Repair Request ADU is multicast to the group and contains the following type of information.

- protocol version
- file identification information
- source IP address
- flags
- missing packet indicator
- missing data (bitmasks or lists)

Receiver scalability across large groups is another important feature of MDP. The key to this capacity is that receivers primarily use NACKs, or missing data repair requests, for feedback to the source. NACK-based reliability sharply reduces receiver requests to the source. Receivers only request data not received by detected gaps in received data and suppress repair requests by tracking duplicate requests within the multicast group. See the recovery cycle section for more discussion of this feature. The missing packet indicator optionally uses a number of different approaches. It can represent missing packets by providing a list of missing packet ids (e.g., counters) or through the transmission of a binary bit mask array representing the received or missed block of packets. In the case of erasure-based parity correction, this segment may contain the maximum number of missing packets and the block id or a set of parity packet bitmasks.

Command ADU Packet

The Command ADU contains the following type of information.

- protocol version
- file identification information
- flags
- cycle duration
- list of positively acknowledged receivers

Command ADU packets are source requests for quick timed delay responses from receivers. Based on the flag settings, the source can optionally request positive acknowledgment (PACK) of specific files or only request for responses if data was not received (NACK). Receivers accordingly respond with associated NACK and PACK responses, however receivers will not send a duplicate

NACK if another receiver has been heard sending the same request within the present recovery cycle. The command packet also provides the mechanism for a return handshake to receivers indicating positive acknowledgement reception. This is done by including in the Command packet, a variable length list of confirmed list of receiver IP addresses from which the source has received a positive acknowledgment. See the recovery cycle and the statistics report packet for a more information.

Statistics Report ADU Packet

The statistic ADU packet provides general statistical summary information from a receiver that has been receiving data from a source. The Statistics Report ADU contains the following type of information.

- protocol version
- file identification information
- source IP address
- flags
- complete files received
- incomplete files
- total packets received
- total number of retransmission requests

Receivers to provide positive receipt of file reception to MDP sources can use the file identification information as a handshake indicator of complete reception.

Transmission Cycle Overview

The MDP source transmission cycle is briefly described as follows. The MDP source enters the initial transmission cycle and multicasts an ID packet specifying the name of the file, its size, and the delay between transmissions. The source begins multicasting the contents of the file using data packets uniformly distributed in time based upon the present transmission rate value (this value can be dynamically changed if a rate-based congestion control algorithm is in use). Each data packet contains a file identifier and a file offset pointer to uniquely identify each packet for receiver reassembly. Upon completion, the source enters a series of recovery cycles to retransmit missing data packets reported by receivers. The source repeats the recovery cycle process until it receives no more requests from receivers or until a timeout expires. A summary state diagram of the source transmission cycle is shown in Figure 2.

```
{send ID}->{send data}-> {recovery cycle} -> {no requests/timeout}
                        ^     |  ^                       |
                        |     v  |                       v
                        | {retransmit data}       {send cmd seq
                        |                          nack request}
                        |                                |
                        |                                v
                        |<---{receiver resp}<-{yes}<-{any response?}
                                                         |
                                                         v
                                                       {no}
                                                         |
```

```
                                                       v
                                                   {finish}
```

Figure 2: Source State Diagram

Recovery Cycle Overview

The following steps detail what happens in the MDP recovery cycle. During the recovery cycle receivers make repair requests by providing an aggregate list of missing packets to the source. This list of requested packets is transmitted within the multicast group. Since the receivers make random delay requests over a backoff window, the probability of receivers sensing duplicate repair requests within multicast group responses is increased.

All packets sent by the source during the recovery cycle contain an EOF flag setting and a recovery cycle flag which marks a transition to a new recovery cycle. To mark the cycle, the source first broadcasts an ID packet, then uses a heartbeat timer setting (e.g., 2 seconds) to trigger successive command packet transmissions for synchronizing receivers. When the receiver detects entry into a new recovery cycle, a random time delayed missing repair request packet response is triggered. Each receiver is allowed only one random time delay request for missing packets within a given recovery cycle. The receiver request for missing packets should not repeat any missing packet requests previously heard from any other receivers during that recovery cycle. The source will immediately retransmit missing packets reported while continuing to listen for additional receiver repair requests. Upon completing retransmission, the source begins a new recovery cycle and sends another ID packet and set of command packets. If no receiver requests are heard during the recovery cycle the source will time out dependent upon on the present frequency settings. The purpose of a source controlled recovery cycle period is to shorten the duration of the cycle period and to increase the turnover frequency of recovery cycles. A higher recovery cycle turnover frequency results in faster file transfer to all receivers. An overview of the source recovery cycle state diagram is shown in Figure 3.

```
{send ID/toggle header flag}->{wait period}-->{no requests/timeout}
               ^                       |                   |
               |                       V                   v
               |               {request heard} <--|   {send cmd}
               |                       |          |        |
               |                       V          |        |
               |<-----------{retransmit data}     |        |
                                                  |        V
                                                  - ---{timed out}
                                                           |
                                                           V
                                                         {End}
```

Figure 3: Source Recovery Cycle State Diagram

In the present design, each receiver tracks delivery of the file in block segments. If the file is larger than this size, it dynamically allocates memory to track the additional data segment. If a receiver determines it is also missing packets at the end of a block segment and is allowed to make requests for missing packets by the source (auto request header flag setting), the receiver enters into a recovery cycle phase as defined above. Upon completion of the recovery cycle, the source resumes transmitting the file where it left off. If the receiver determines the source has transitioned to a new

data segment (as defined above) it will reenter the recovery cycle phase to request any missing packets if they still exist. This mode of operation allows for data repair cycles to occur at defined intervals during the initial data transmission rather than requiring multiple passes upon one complete transmission of the file. The purpose of this feature is to regulate the source from advancing too far ahead of receivers requiring repair packets.

When servicing a missing data repair request, the sources automatically multicast all data packets requested. Upon fulfilling all requests the source will send another ID packet and toggle the recovery cycle flag. The recovery cycle flag indicates to all receivers the beginning of a new recovery cycle. The recovery cycle time duration is determined by a timer value for the heartbeat interval (e.g., 2 secs). The source continues the recovery cycle process until the source completes one cycle without any receiver repair requests being received. If at this time the source has completed the file transmission, the source will send a periodic sequence of command packets with the NACK flag set. Upon hearing this command packet, receivers that have not received a complete file are designed to do a short time delayed response to the source to keep it in the recovery cycle. Once again, a receiver will not make a response if it had previously heard a similar receiver response. The source only needs to hear one response before starting back into the recovery cycle. An overview of the receiver recovery cycle is shown in Figure 4.

```
{Packet contains EOF} -> {Toggled Recovery Flag?} ->{Initiate time
        ^                                             delay response}
        |                                                   |
        |                                                   v
        |                                           {Listen for other
        |                                            receiver requests}
        |                                                   |
        |                                                   v
        |< -{Send non-duplicate}<-{missing data} <-- {Timeout}
        |    {repair request}                               |
        |                                                   |
        |                                                   |
        |                                                   v
        |                                           {file completed}
        |                                                   |
        |                                                   v
        |<---{send stat report}<---- {no} <-- {source heard PACK?}
                                                            |
                                                          {yes}
                                                            |
                                                            v
                                                    {End of cycle}
```

Figure 4: Receiver Recovery Cycle State Diagram

An optional operational mode is available in which the source can request positive acknowledgment of complete file reception from receivers, the source will set the PACK flag in all outgoing packets. Upon file completion, receivers will multicast a random time delay stat packet. For each command packet received from the source, each receiver will continue to send a stat packet (at a heartbeat interval) until it has timed out or has received a command packet from the source acknowledging receipt of receiver’s stat message. This optional mode of operation is not recommended when large group membership is anticipated, due to the corresponding increase in multicast message traffic. It can, however, provide list-based file receipt assurance for particular members when desired.

To improve achievable throughput, the source may initiate another file transfer while being in the recovery cycle of another. The transmission rate remains constant since the new file transfer process and existing recovery cycle are performed asynchronously. The source always responds first to receiver requests then resumes new file transfer. As detailed above in the source recovery cycle, anytime the source's packet transmission changes to a new data segment being tracked by the receiver, the receiver will enter into a recovery cycle to request any missing packets in the old segment.

Future Work and Design Issues

While the present design has been through limited Mbone testing and has been shown to work effectively and efficiently for a number of applications, there remain are number of design issues which the authors envision will continue to evolve. One of these issues is future approaches to flow control and congestion avoidance within a multicast group environment. We feel this is a general problem and not unique to MDP. While effective reactive flow control in a multicast environment remains a complex technical design issue, there are basic rate control features in the present design that can be controlled dynamically.

In continuous transmit mode, the source can optionally adjust the data transfer rate by monitoring feedback of total packet retransmission requests from receivers as compared to total packets sent for each file. The authors are aware that future design modifications are likely to occur here since many important issues remain unresolved concerning reliable multicast reactive flow control for WAN environments. We are exploring modifications to the protocol framework in this area. Nonetheless, in many instances, source rate control can work effectively (e.g., combination with a resource reservation protocol). We recognize the difficulty of managing a large group session around single or small populations of faulty or poorly performing receivers operating below a desired group throughput threshold. As is presently done with many other Mbone applications (e.g., compressed video, audio), we recommend that MDP/IMM users pay close attention to initial rate settings of their sources. To prevent accidental poor practice, reasonable lower and upper rate limit settings and default values are used within the distributed software implementation release. In summary, MDP provides some optional rate adaption capability based upon negative acknowledgements experienced within the recovery cycle. There are two reasons we caution against protracted use of this technique: packet loss does not always indicate congestion and the delayed aggregation of repair messages delays the statistical feedback to the source.

On the upside, NACK aggregation and duplicate request suppression at receivers keeps reliable control loop traffic somewhat minimized during bulk data transfer. This simple approach can be quite effective for a number of non-real-time bulk file transfer applications.

We envision future, potential advantage in applying this protocol framework in combination with a reservation protocol (e.g.,RSVP [6]) and future integrated or differential services capabilities. The source rate control setting can be reflective of the bandwidth reserved and protocol timers can be better tuned to operate within average or upper bound delay expectations. Proactive flow control is supported through Integrated Services Architecture (ISA) components and the protocol

does not have to surrender its message reduction efficiency for the sake of reactive feedback control. Such a coupling is not required for protocol correctness, but operation in conjunction with underlying ISA capabilities can support better overall performance.

In addition, for high error rate and asymmetric network channels the adaptation of MDP to a hybrid reliable multicast dissemination scheme using both forward error correction and retransmission is presently under design and consideration. Work on MDP version 2 is underway at NRL to look at this issue and develop more advanced solutions beyond the MDPv1 framework presented here.

Suggested Usage

As mentioned, the present MDP framework is seen as useful for the reliable bulk file transfer over generic IP multicast services. It is not the intention of the authors to suggest it is suitable for supporting all envisioned multicast reliability requirements, but rather it provides a simple framework for multicast file dissemination applications with a degree of concern for network traffic implosion. As previously described, over several years IMM has been successfully demonstrated within the MBone for reliable bulk data dissemination applications, including weather satellite compressed imagery updates servicing a large group of receivers.

In addition, this framework approach has some design features that make it attractive for bulk transfer in future asymmetric network applications. The multipass repair cycles allow receiver group members to better aggregate and minimize duplicate repair requests with looser timing estimation and windowing requirements than approaches designed for smaller messaging and real-time interaction. A source-only repair approach with unicast feedback may also make technical sense in asymmetric networks. Asymmetric architectures supporting multicast delivery are likely to make an important portion of the future Internet structure (e.g., DBS/cable/PSTN hybrids) and efficient, reliable bulk data transfer will be an important capability for servicing large groups of subscribed receivers.

References

[1] W. Dang. "Reliable File Transfer in the Multicast Domain". Technical Report. August 1993

[2] S. Deering. "Host Extensions for IP Multicasting". Internet RFC 1112, August 1989.

[3] J. Chang and N. Maxemchuk. "Reliable Broadcast Protocols". ACM Transactions on

[4] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang. "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing". In Proc. ACM SIGCOMM, August 1995.

[5] D. Clark and D. Tennenhouse. "Architectural Considerations for a New Generation of Protocols". In Proc. ACM SIGCOMM, pages 201--208,

September 1990.

[6] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. "RSVP: A New Resource ReSerVation Protocol". IEEE Network Magazine, pages 8--18, September 1993.

[7] J. Macker, M. Corson, E. Klinker, "Reliable Multicast Data Delivery for Military Internetworking". IEEE MILCOM 96 Proceedings, pages 399-403, October 1996.